

## Applicant Initiated Interview Request Form

Application No.: 10/749,757 First Named Applicant: Nikolai G. Nikolov  
 Examiner: Rutten, James D. Art Unit: 2192 Status of Application: pending

**Tentative Participants:** Spencer K. Hunter, Law Clerk for

- (1) Gregory D. Caldwell, undersigned attorney of record (2) Examiner Rutten  
 (3) \_\_\_\_\_ (4) \_\_\_\_\_

Proposed Date of Interview: Wed, May 5, 2010 Proposed Time: 9am PST and 12pm EST AM/PM

**Type of Interview Requested:**

- (1) ☒ Telephonic (2) ☐ Personal (3) ☐ Video Conference

Exhibit To Be Shown or Demonstrated: ☐ YES ☐ NO

If yes, provide brief description: \_\_\_\_\_

### Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) _____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(2) _____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(3) _____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(4) _____	_____	_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Continuation Sheet Attached

**Brief Description of Argument to be Presented:**

Discuss proposed Examiner's amendment to place case in immediate condition for allowance.

An interview was conducted on the above-identified application on \_\_\_\_\_.  
**NOTE:** This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).  
 This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

/Gregory D. Caldwell/  
 Applicant/Applicant's Representative Signature  
Gregory D. Caldwell  
 Typed/Printed Name of Applicant or Representative  
39926

\_\_\_\_\_  
 Examiner/SPE Signature

\_\_\_\_\_  
 Registration Number, if applicable

**FOR EXAMINER'S CONSIDERATION DURING EXAMINER'S INTERVIEW**  
**\*\* DO NOT ENTER \*\***

**SECTION I—CLAIMS**

**Listing of Claims:**

40. (Currently amended) A method for modifying an application to provide functionality for tracing a program flow of the application at a user-configurable level of granularity specified via a Graphical User Interface presented at an end-user device, the method comprising:

reading program code from memory and processing said program code with one or more processors to perform the following:

presenting to the end-user device via the Graphical User Interface, providing a user with options for modifying the an application's bytecode by injecting tracing and debugging operations into the application's bytecode at the user-configurable level of granularity specified via the Graphical User Interface, wherein the application is, said application composed of a plurality of archive files, the said archive files having respective class files, and the said respective class files having respective methods, and wherein the said options for modifying the application's bytecode includes:

including one or more of the following: i)

modifying bytecode of only one classfile a subset of a plurality methods from which the application is composed, the subset of the plurality of methods selected from one or more class files in one or more within any one of said archive files composing the application's bytecode as specified via the Graphical User Interface presented to the end-user device, wherein the modified subset of the plurality of methods specified provides the user-configurable level of granularity by providing the

**FOR EXAMINER'S CONSIDERATION DURING EXAMINER'S INTERVIEW**  
**\*\* DO NOT ENTER \*\***

functionality for tracing the program flow of the application through only the subset of the plurality of methods specified via the tracing and debugging operations injected into the subset of the plurality of methods specified;

~~ii) modifying bytecode of only one method within only one of said archive files' respective~~  
~~classfiles;~~

~~modifying bytecode of said application in accordance with said user's selection of one of said~~  
~~options;~~

executing the said application in an object oriented runtime frame work, wherein said executing the application includes ~~including~~ processing a portion of the said application's bytecode that was modified in accordance with the user-configurable level of granularity specified via the Graphical User Interface; ~~and said user's selection of one or more of said options;~~  
~~and,~~

~~presenting to said user an~~ output generated from execution of the said portion of the said application's bytecode that was modified to the end-user device via the Graphical User Interface.

**FOR EXAMINER'S CONSIDERATION DURING EXAMINER'S INTERVIEW**  
**\*\* DO NOT ENTER \*\***

Applicant's Specification teaches:

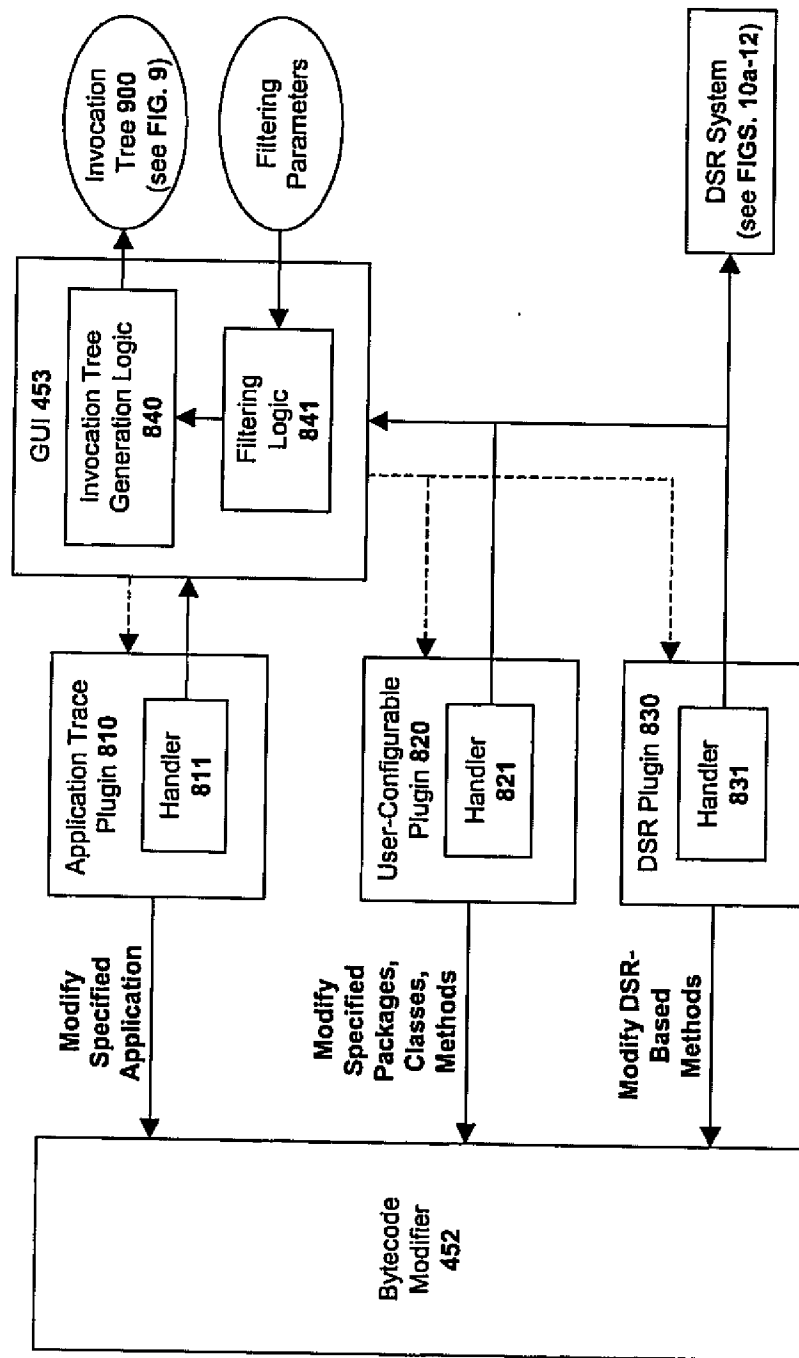
[0114] As described above, the application tracing plugin 810 instructs the bytecode modifier to modify all of an application's methods. While this may be sufficient for tracing/debugging a relatively small application, a higher level of precision may be desirable, particularly when working with large enterprise applications.

[0115] Thus, in contrast to the application tracing plugin 810 which causes the bytecode modifier 452 to modify all of the methods within a particular application, the user-configurable plugin 820 illustrated in Figure 8 provides a finer level of granularity for tracing program flow. An "application" may be built from a plurality of packages (typically \*.jar files in a Java environment); each package may be built from a plurality of classes (i.e., class files); and each class include a plurality of methods. As indicated in Figure 8, the user-configurable plugin 810 allows the end-user to identify specific packages, classes and/or individual methods to be modified by the bytecode modifier 452, thereby providing significantly greater precision for tracing and debugging operations. By way of example, if a coding problem is isolated to within a specific package, then only that package need be modified. Similarly, if the problem can be isolated to within a particular class or method, then only that class/method need be modified. In one embodiment, the different packages, classes and/or methods are selected and modified via one of the interfaces described below with respect to Figures 19a-e.

[0116] The method timing data and/or other method-related information may then be displayed within a method invocation tree 900 similar to that illustrated in Figure 9a. For example, an output handler 821 associated with the user-configurable plugin 820 may be designed to provide the method-related information to the invocation tree generation logic 840. **Thus, when the user-configurable plugin 820 is employed (as opposed to the application tracing plugin 810), the method invocation tree does not include entries for all of the methods of an application. Rather, it only includes entries for methods within the particular package or class, or the individual methods selected by the end-user.**

[0117] In addition, in one embodiment of the invention, a distributed statistical records ("DSR") plugin 830 is employed to collect statistical data related to program execution across application servers, databases and/or external systems. Several specific examples of DSR tracing will now be described with respect to Figures 10a-12, which shows how certain critical entry/exit methods may be tracked to collect statistical data.

**FOR EXAMINER'S CONSIDERATION DURING EXAMINER'S INTERVIEW**  
**\*\* DO NOT ENTER \*\***



**FIG. 8**